CMPE 124 Lab 10: Shift registers

Leonardo Blas

***Abstract*—The purpose of this lab project is to implement a shift register circuit via D flip flops.**

I. Introduction

This lab project is based on the implementation of a 74LS163 counter, a logical circuit that tracks time through 4 bits.
Furthermore, we also construct an 8-bit shift register, using D flip flops, where we use our counter's output as input.

II. Design Methodology

For this lab project, we use the 74LS163 4-bit counter. This 4-bits counter requires not only a voltage source—to function—and a clock—to provide it with time input—but also a binary switch, to reset it and ensure the wave functions it outputs are correct. Additionally, although we use a resistor, its value can be changed or ignored; our 1kΩ resistor is part of the circuit to demonstrate that we are not focusing on the input voltage, but on the H and L states instead.
Furthermore, we implement an 8-bit shift register via D flip flops and use Qc—our counter's second most significant bit—as input for this shift register.
Finally, once our shift register has been implemented, we can simulate that it receives data, through our counter's input, and then we could design a system—using logic gates—that detects the ASCII values of "A" and "G".

A. Parts List

- Clock
- 5V Source
- 74LS163
- Binary switch
- Resistor
- NOT gates
- D flip flops

B. Truth Tables

State	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
q_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
q_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
q_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
q_3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 1. Truth table for our 4-bit counter.

C. Schematics

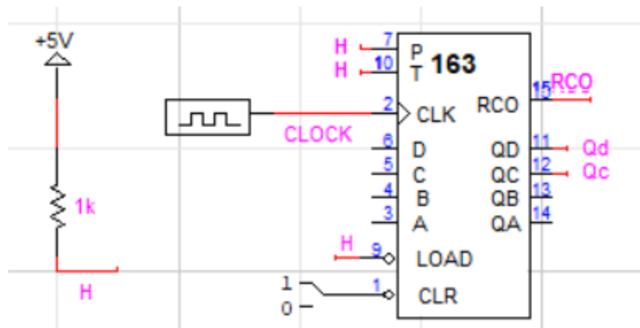

Figure 2. 74LS163 counter circuit.

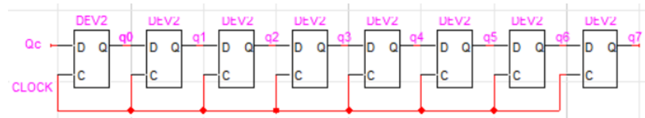

Figure 3. 8-bits shift register.

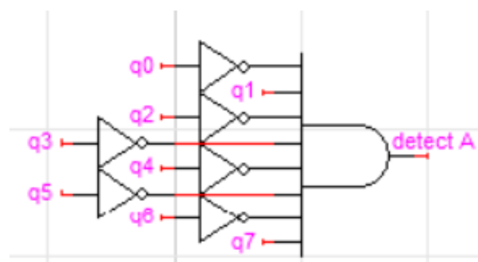

Figure 4. Branch to detect the ASCII value of "A".

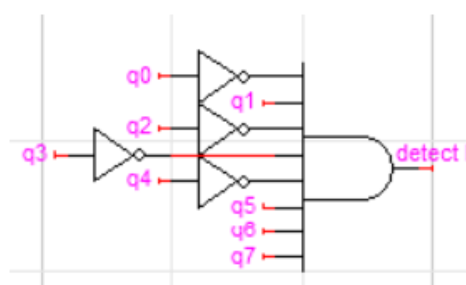

Figure 5. Branch to detect the ASCII value of "B".

III. Testing Procedures

1. Create a counter, as pictured in Figure 3.
2. Set up a the 8-bit register, as shown above.
3. Create the logic systems to detect the ASCII values of "A" and "B," as shown above.
4. Reset the switches and run the simulation.

5. Record each circuit’s waveforms.

IV. TESTING RESULTS

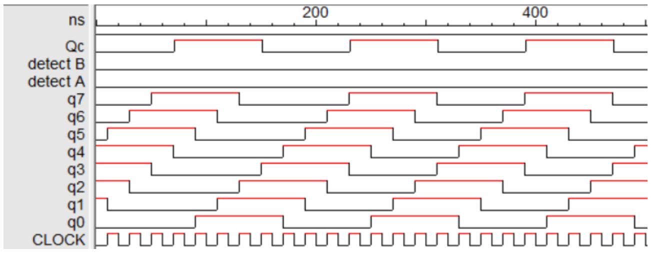

Figure 6. The 8-bit shift register’s and the “A” and “B” ASCII detectors’ waveforms.

V. CONCLUSION

In conclusion, this lab taught us that we can cascade several flip flops together in order to store binary data and later identify specific numbers within a binary string of data. Furthermore, notice that our “A” and “B” ASCII detectors were not triggered (see the waveforms in Figure 6). We could argue that our detectors weren’t triggered due to the uniformity of the data that was being fed to them (derived from the 74LS163 counter), and that the “substrings” of data that were meant to trigger our detectors—01000001 and 01000111—were nowhere to be found in our data stream (we could tell this by seeing the waveforms). Ultimately, it’s important to say that while we have found yet another use for the 74LS163 clock-counter circuit we’ve been typically using in most of our projects—this time we use it to feed information to our shift register—the concept of flip flops is still fairly new to us, and this was a good opportunity to become exposed to the hardware components that allows computers to operate as they do.